UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/964,807	09/28/2001	Stephan J. Jourdan	2207/12004	2860

23838 7590 10/27/2008

KENYON & KENYON LLP
1500 K STREET N.W.
SUITE 700
WASHINGTON, DC 20005

EXAMINER	
COLEMAN, ERIC	

ART UNIT	PAPER NUMBER
2183	

MAIL DATE	DELIVERY MODE
10/27/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	09/964,807	JOURDAN, STEPHAN J.	
	Examiner	Art Unit	
	Eric Coleman	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 7/11/08.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 5-28 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 5-28 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date. _____
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____

DETAILED ACTION

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 5-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Keller (patent No. 6,622,237) in view of Feiste (patent No. 6,349,382).

Keller taught the invention substantially as claimed including a data processing ("DP") system comprising (as per claim 5):

A scheduling method for a load instruction comprising: if a new load microinstruction is admitted, predicting whether collision occurs between load microinstruction and an older microinstruction (col. 10, lines 35-65 show that a STLF, store to load forwarding, col. 2, lines 14-27) predictor makes a prediction for each load instruction (new load) based on past executions whether it will interfere or collide with a store operation. Col. 10, line 61-col. 11, line 8 shows that if embodiment of focus in the disclosure is where the store is older than the load), if a collision is detected whether data for the older store microinstruction is available, if data for the older store is not available, storing the load microinstruction in a scheduler with a marker indicating that scheduling of the load microinstruction is to be deferred [col. 12, lines 7-22 show that a PA buffer is used to detect stores that interfere of collide with loads. this section also shows that when a collision (a match by a PA with loads). This section also shows that

when a collision (a match with the PA buffer) is found and the corresponding operation is a load that is younger than a store, then the scheduler entry is defined that indicates the load is retried, that is the load is not to be executed (deferred) since it has a dependency on an older store for which the data is not available and will be retried later. Markers (figure 3, element 6 R and T). are used to convey that a load is to be retried and that a load is dependent on an older store as shown in col. 11, lines 25-31.

Keller did not expressly detail (claim 5) that the determination determined whether data was available in a store unit. Feiste however taught if a collision is detected between a store and load a store unit it is determined whether the data needed by a load from an older store is available in a store unit for forwarding the data to the load (e.g., see col. 5, lines 14-67).

It would have bee obvious to one of ordinary skill in the DP art to combine the teachings of Keller and Feiste. Both references were directed toward the problems of scheduling instructions including load instruction and store instructions allowing out of order execution. One of ordinary skill would have been motivated to incorporate the Feiste teachings of checking a store unit for the data needed by a load at least to reduced time taken to process the load instruction. Also the incorporation of the Feiste teachings would have yielded predictable results.

As per claim 6, Keller taught storing a scheduler entry identifier (fig. 3, element 66, SID) of the older store with the load microinstruction [col. 11, lines 22-25 that show the SID field in the scheduler entries indicate the store (older store) operation interfered (collided) with].

As per claim 7, Keller taught scheduling the load microinstruction for execution after the marker is cleared [col. 121, lines 44-46 show that scheduling of a load for execution (it is still placed in the scheduling buffer when not executed as shown above) is inhibited until after the store is scheduled for execution. This means that the load entry in the scheduler has markers preventing execution, as discussed previously, will clear the markers so that execution proceeds].

As per claim 8, Keller taught a system an method that prevents conflicts in instructions namely between a load and store and storing each instruction operation until the dependencies noted for the instruction operation have been satisfied before scheduling the instruction(e.g., see col. 7,lines 6-53). Therefore one of ordinary skill would have been motivated to ensure proper processing each instruction to be processed which would have included scheduling other instructions dependent upon the load microinstruction to execute after the load microinstruction executes.

As per claim 9, Keller taught deferring the scheduling of a each instruction including a load when a conflict occurred because a store which the load was dependent has not completed (e.g. see col. 7, lines13-52). Therefore one of ordinary skill would have been motivated to defer the scheduling of other instructions dependent upon the load microinstruction when scheduling of the load microinstruction is deferred at least to similarly ensure that the correct data was available when the other instruction were to access the load data.

As per claim 13, Keller taught An execution unit for a processing agent comprising a scheduler (figure 1, element 36) operating according to the method of claim 5, a register file(figure 1, element 38A), and a plurality of execution modules(figure 1, elements 40A,40B), wherein the scheduler, the register file and the execution modules each are coupled to a common communication bus [figure 1 shows that the scheduler, register files and execution modules have a bus that begins at the scheduler, passes through the register file, and couples the execution modules together.

As per claim 14, Keller taught A scheduling method comprising: predicting wherher a new load microinstruction (e.g., see younger loads in col. 11, lines 45-49) collides with a first previously received store microinstruction [(col. 10, lines 35-55 show that a STLF store to load forwarding, col. 2, lines 14-27) predictor makes a prediction for each load instruction (new load) based on past executions whether it will interfere of collide with a store operation col. 10, lines 61-col. 11, line 8 shows that the embodiment of focus in the disclosure is where the store is older than the load] when the new load microinstruction is admitted to a scheduler, if a collision is detected determining whether data for the older store microinstruction is available in a store unit if data for the older store is not available, storing the load microinstruction in the scheduler with a dependency pointer to a second previously received store microinstruction [col 12, lines 7-22 show that a PA buffer is used to detect stores that interfere or collide with loads. this Section also shows that when a collision (a match by the PA buffer is found and the corresponding operation is a load that is younger than a store, then a scheduler entry is defined that indicates the load is to be retried, that is the load is not to be executed

(deferred since it has a dependency on an older store for which the data is not available and will be retried later. Markers or pointers (figure 3, element 6, R and T) are used to convey that the load is to be retried and that a load is dependent on an older store as shown in column 11, lines 25-31. Since the terms first and second are merely names and may in fact name the same element, the Examiner is taking the first and second microinstructions to be the same microinstruction].

Keller did not expressly detail (claim 14) a determination was made determining whether data for the load was available in a store unit. Feiste however taught if a collision is detected between a store and load a store unit it is determined whether the data needed by the load from the older store is available in a store unit for forwarding the data to the load (e.g., see col. 5, lines 14-67).

It would have bee obvious to one of ordinary skill in the DP art to combine the teachings of Keller and Feiste. Both references were directed toward the problems of scheduling instructions including load instruction and store instructions allowing out of order execution. One of ordinary skill would have been motivated to incorporate the Feiste teachings of checking a store unit for the data needed by a load at least to reduced time taken to process the load instruction. Also the incorporation of the Feiste teachings would have yielded predictable results.

As per claim 15, Keller taught scheduling the load instruction for execution after the dependency pointer cleared [col. 12, lines 44-46 show that scheduling of a load for execution (it is still placed in the scheduling buffer when not executed as shown above)

Is inhibited until after the store is scheduled for execution. This means that the load entry in the scheduler has markers preventing execution, as discussed previously, will clear the markers so that execution proceeds][col. 7 lines 13-52 show deferring the scheduling of each instruction until dependencies were satisfied].

As per claim 16, Keller taught deferring the scheduling of each instruction until dependencies were satisfied (including a load instruction when a conflict occurred because a store which the load was dependent has not completed (e.g. see col. 7, lines13-52). Therefore one of ordinary skill would have been motivated to schedule other instruction dependent upon the load instruction to execute after the load instruction executes.

As per claim 17, Keller taught deferring the scheduling of each instruction until dependencies were satisfied (including a load instruction when a conflict occurred because a store which the load was dependent has not completed (e.g. see col. 7, lines13-52). Therefore one of ordinary skill would have been motivated to defer the scheduling of other instructions dependent upon the load instruction when scheduling of the load instruction is deferred.

As per claim 21, Keller taught a scheduler (figure 1, element 36) operating according to the method of claim 14, a register file (figure 1, element 38A) ; and a plurality of execution modules (figure 1, elements 40A, 40B), Wherein the scheduler the register file and the execution modules each are coupled to a common communication bus.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 22-28 are rejected under 35 U.S.C. 102(e) as being anticipated by Keller (patent No. 6,622,237).

As per claim 22, Keller taught a dependency management comprising, upon execution of a STD uop: comparing an identifier of the STD microinstruction to dependency pointers of other microinstructions stored by a scheduler, (col. 12, lines 7-22 show that PA buffer entries (identifiers) are compared for store (STD) and load operations or microinstructions (uops) and clearing any dependency pointers that match the identifier[(figure 3, element 6, R and T) are used to convey that a load is to be retried and that a load is dependent on an older store as shown in column 11, lines 25-31. Column 12, lines 44-46 show that scheduling of a load for execution (it is still placed in the scheduling buffer when not executed as shown above) is inhibited until after the store is scheduled for execution. This means that the load entry in the scheduler that has markers preventing execution will clear the markers so that execution proceeds].

As per claim 23, Keller taught the identifier represents a location in a store unit where data responsive to the STD microinstruction is stored Column 12, lines 7-22 and figure 3 show that each PA buffer entry (identifier) corresponds to a scheduler entry and thus represents a location in the scheduler that contains a uop].

As per claim 24, Keller taught he identifier represents a location in a store unit where data responsive to the STD microinstruction is stored (column 11, line 67-col.12, line 1 show that a physical address (just as is stored in the PA buffer and thus is the same identifier) is provided to the store queue for storage). Since the identifier (physical address is in itself a location and is stored in this store unit, the claim is met.

As per claim 25, Keller taught the STD microinstruction causes a transfer of data to a store buffer when executed (e.g., see col. 10, lines 34-46).

As per claim 26, Keller taught the STD microinstruction is paired with a STA microinstruction that, when executed causes calculation of a store address [(see how the operands added produce virtual address of a store operation in col. 11, lines 61-67, and col. 12, lines 1-6].

As per claim 27,Keller taught a dependency management method comprising: decoding a store instruction as a plurality of microinstructions, including a STA and a corresponding STD microinstruction (e.g., see decoding of store and load instruction in col. 5, lines 28-56), decoding a load instruction as at least one microinstruction, when a dependency between the STA microinstruction and the load microinstruction occurs (see the dependency between the load and store instructions in col. 10, lines 34-46), deferring scheduling of the load microinstruction until after the corresponding STD

microinstruction executes (see load instruction does not get scheduled prior to the store in col. 10, lines 40-46 , see also the retry of the load on matching condition with the store in col. 12, lines 11-22).

As per claim 28 Keller taught detecting the dependency between STA microinstruction (see store instruction) and the load microinstruction (see load instruction) by comparing an identifier (see match found of physical address col. 12, lines 10-22) of the STD microinstruction to dependency pointers of other microinstructions stored by a scheduler (66,70), and clearing any dependency pointers that match the identifier (see the rejection state being reset).

Claim Rejections - 35 USC § 103

Claims 10-12 and 18-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Keller and Feiste as applied to claim 5 above, and further in view of Abramson (patent No. 5,898,854) and Hennessy (Computer Organization and Design).

As per claim 10, Keller and Feiste taught the scheduling method of claim 5 (as discussed above). Keller and Feiste did not expressly detail wherein the store microinstruction is one of a plurality of microinstructions representing a store instruction, wherein a first microinstruction is to transfer data to a store unit and a second microinstruction is to calculate an address of the store instruction. Abramson, however taught (e.g., see col. 6, lines 12-15) that the store instruction is made up of address (calculate address) and store data (transfer data) operations or microinstructions. Hennessy taught in the introduction to section 5.5 on pages 399-400 that microinstructions provide method of specifyijng control that makes understanding and

design of a system easier since complex instructions are broken up into smaller and simpler operations. The ability to make system design an comprehension easier would have motivated one of ordinary skill in the art to use microinstructions to specify as many simple operations as possible, specifically the store address and store data microinstruction of Keller is broken up further into store address and store data microinstructions of Abramson. , Keller taught the store microinstruction is a part of a plurality of microinstructions representing a store instruction, wherein a first of the plurality to transfer data to the store unit and a second of the plurality is the older store microinstruction(e.g., see col. 11, lines 1-52), which is to calculate an address of the store instruction.

It would have been obvious to one of ordinary skill in the DP art at the time of the invention to modify the design of Keller to use the store address and store data microinstructions for the store instructions so that design an understanding is easier as taught by Hennessy.

In regard to claim 11, Keller and Feiste in view of Abramson and Hennessy disclose the scheduling method of claim 10, further comprising the marker of the load microinstruction after the first store microinstruction executes. Keller at Column 12 , lines 44-46 show that scheduling of a load for execution (it is still place in the scheduling buffer when executed as shown above) is inhibited until after the store is scheduled for execution. This means that the load entry in the scheduler that has markers preventing execution, as discussed previously, will clear the markers one the store is completed and the dependency is removed so that execution proceeds.

As per claim 12, Keller in view of Abramson and Hennessy disclosed the scheduling method of claim 10 wherein the prediction determines a collision between the load microinstruction and the second store microinstruction. As shown above in the sections cited , the dependency between the load and store instructions stems from the data not being ready and therefore the second microinstructions, which directs the transfer data of data, is the microinstruction that a collision with the load is detected and predicted for.

As per claim 18, Keller taught the scheduling method of claim 5, Keller does not expressly detail wherein the store microinstruction is one of a plurality of microinstructions representing a store instruction, wherein a first microsinstruction is to transfer data to a store unit and a second microinstruction is to calculate an address of the store instruction. Abramson has taught in column 6, lines 12-15, that the store instruction is made up of store address (calculate address) and store data (transfer data) operations or microinstructions Hennessy taught in the introduction to section 5.5 pages 399-400 that microinstructions provide method of specifyijng control that makes understanding and design of a system easier since complex instructions are broken up into smaller and simpler operations. The ability to make system design an comprehension easier would have motivated one of ordinary skill in the art to use microinstructions to specify as many simple operations as possible, specifically the store address and store data microinstruction of Keller is broken up further into store address and store data microinstructions of Abramson. Also Keller included a first

plurality (see load instructions) and a second plurality (store instructions , older stores in col. 11, line 52 of Keller).

It would have been obvious to one of ordinary skill in the DP art at the time of the invention to modify the design of Keller to use the store address and store data microinstructions for the store instructions so that design an understanding is easier as taught by Hennessy. Also the incorporation of the Hennesy teaching would have yielded predictable results.

As per claim 19 Keller in view of Abramson and Hennessy taught the scheduling method of claim 18, further comprising clearing the dependency pointer of the load microinstruction after the store microinstruction executes. Keller at Column 12, lines 44-46 show that scheduling of a load for execution (it is still place in the scheduling buffer when executed as shown above) is inhibited until after the store is scheduled for execution. This means that the load entry in the scheduler that has markers preventing execution, as discussed previously, will clear the markers one the store is completed and the dependency is removed so that execution proceeds (see dependencies for dependency unit in col. 10, lines 30-55 of Keller)

As per claim 20, Keller in view of Abramson and Hennessy disclosed the scheduling method of claim 18 wherein the prediction determines a collision between the load microinstruction and the second store microinstruction. As shown above in the sections cited , the dependency between the load and store instructions stems from the data not being ready and therefore the second microinstructions, which directs the

transfer data of data, is the microinstruction that a collision with the load is detected and predicted for.

Response to Arguments

Applicant's arguments filed 7/11/08 have been fully considered but they are not persuasive.

The rejections are maintained as set forth in the last office action and presented above. The portions of the last office action that were incorporated by reference from a previous office action is also included above.

The change in scope of the amended claims has necessitated a new search.

The applicant argues in substance that a newly added feature (if a load/store collision is detected, it is determined whether the data is available in a store unit) is allegedly not taught by Keller or the other cited prior art. As this is a newly added feature additional art (Feiste) is used to meet the limitation (see rejections above).

The application also argues the claims specifically deal with the STA and STD uop pair of a store instruction and alleges that the cited sections in outstanding rejection does not teach of disclose specific features related to the these claims. This argument is vague and does not specify what the difference between the prior art and claims includes. The outstanding rejection discusses the inclusion of the STA and STD uop pair therefore the arguments are not persuasive.

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Olson (patent No. 5,878242) disclosed a method and system for forwarding instructions in a processor with increased forwarding probability (e.g., see abstract).

Moore (patent 6,938,148) disclosed a system managing load and store operations using a storage management unit with data flow architecture (e.g., see abstract).

Rodgers (patent No. 5,636,374) disclosed a method and apparatus for performing operations based on addresses of microinstructions (e.g., see abstract and col. 5, line 1-col. 6, line 67).

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Eric Coleman whose telephone number is (571) 272-4163. The examiner can normally be reached on Monday-Thursday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

EC
/Eric Coleman/
Primary Examiner, Art Unit 2183